Rješavanje zadataka iz statistike za srednje škole pomoću programskog jezika *R*

Marina Ninčević[1]

Sažetak. U članku se za pogodno odabrane primjere iz statistike daju detaljna rješenja popraćena potrebnim kodom u programskom jeziku *R*. Pomoću njega se na jednostavan način mogu crtati različiti grafički prikazi podataka, provjeravati rezultati i po potrebi elegantno raspolagati većim brojem podataka. Primjeri pripadaju gradivu nastavnog programa za prirodoslovno-matematičke gimnazije kao i izbornoj nastavi matematike za opće gimnazije. Temeljeni su na višegodišnjem iskustvu držanja vježbi iz kolegija Statistika i Statistički praktikum 1 na Matematičkom odsjeku PMF-a na kojima se seminari i rad u praktikumu izvode u *R*-u.

1. Uvod

R je programski jezik pogodan za obradu podataka i njihovu grafičku reprezentaciju. Njegovi tvorci su R. Ihaka i R. Gentleman sa Sveučilišta u Aucklandu, Novi Zeland. Program je djelomično dobio ime po prvim slovima imena autora. *R* je potpuno besplatan i dostupan za operacijske sustave Unix, Windows i Mac OS na http://www.r-project.org. U drugom poglavlju ovog članka navodi se samo najosnovnije potrebno za prvo susretanje s *R*-om, a poglavlja koja slijede sadrže zadatke s prijedlogom objašnjenja za učenike srednjih škola. Sve funkcije iz *R*-a koje su korištene za grafički prikaz i analizu podataka iz danih zadataka detaljno su objašnjene zajedno s osnovama statistike koje ti zadatci ilustriraju, tako da mogu poslužiti kao uvodni primjeri.

2. Unošenje podataka i pozivanje funkcija u *R*-u

Argumenti su kod pozivanja funkcija u *R*-u navedeni unutar običnih zagrada, na primjer *log(x)* (funkcija koja vraća vrijednost logaritamske funkcije s bazom e u točki *x*).

[1]Marina Ninčević, PMF – Matematički odsjek, Sveučilište u Zagrebu

Najčešće korištena funkcija u danim primjerima je „concatenate" za generiranje vektora (nizova podataka) koja se poziva naredbom *c(...)*, gdje unutar zagrada pišemo podatke odvojene zarezom. Na primjer, *x = c(1, 7, 3)* stvara niz *x* koji sadrži 3 podatka (prvi je 1, drugi 7, a treći 3). Podatcima unutar vektora pristupamo imenom niza nakon kojeg stavljamo redni broj podatka unutar uglatih zagrada. Evo kako se ispisuje drugi element iz vektora *x*:

```
> x = c(1,7,3)
> x[2]
[1] 7
```

Možemo uočiti da se naredbe upisuju u retcima koji započinju znakom >, a rezultati ispisuju u sljedećem retku nakon znaka [1].

Na definiranim nizovima podataka možemo primijeniti razne funkcije od kojih su za statističku analizu često korisne sljedeće: *length(x)* – vraća duljinu vektora *x*, *sum(x)* – vraća sumu svih elemenata vektora *x*. Za gore definirani vektor u *R*-u se dobiju sljedeći rezultati:

```
> length(x)
[1] 3
> sum(x)
[1] 11
```

Dakle, duljina vektora *x* je 3, suma svih elemenata je 11.

Osim zadavanja argumenata pozicijom, specifičnost *R*-a je u tome da se argumenti mogu zadavati i imenom. Većina argumenata ima pretpostavljene vrijednosti koje se koriste (ako drugačije ne zadamo). Na primjer, *log(x, base = 2)* vraća logaritam od *x* s bazom 2, gdje se baza zadaje argumentom *base*. Pretpostavljena vrijednost argumenta *base* je Eulerov broj *e*.

Detalje o svakoj funkciji, kao i popis mogućih argumenata, možemo dobiti naredbom *help*, na primjer *help(log)* ili jednostavnije sa *?log*. Iz preglednika help-a izlazi se klikom na tipku *q*.

Ako želimo prekinuti nedovršenu naredbu i umjesto znaka + dobiti uobičajeni početni znak retka s naredbama >, potrebno je stisnuti *Ctrl+C*.

Iz programa izlazimo sa:

```
> quit()
```

3. Grafičko prikazivanje podataka u *R*-u

Prikupljene podatke možemo prikazivati raznim grafovima, ovisno o tome o kakvim se podatcima radi i što o njima želimo zaključiti. Među uobičajenim prikazima podataka su linijski dijagram, kružni dijagram i stupčasti dijagram. U *R*-u postoji dosta unaprijed pripremljenih funkcija potrebnih za grafičko prikazivanje podataka. Neke od njih su *plot*, *pie* i *barplot.*

Zadatak 1. Sljedećom tablicom dane su površine nacionalnih parkova u Hrvatskoj:

Nacionalni park	Površina (ha)
Plitvička jezera	19 479
Paklenica	3 617
Risnjak	3 198
Mljet	3 100
Kornati	6 900
Brijuni	3 635
Krka	14 200

Podatke iz tablice prikažite u obliku:

(a) linijskog dijagrama,

(b) kružnog dijagrama,

(c) stupčastog dijagrama.

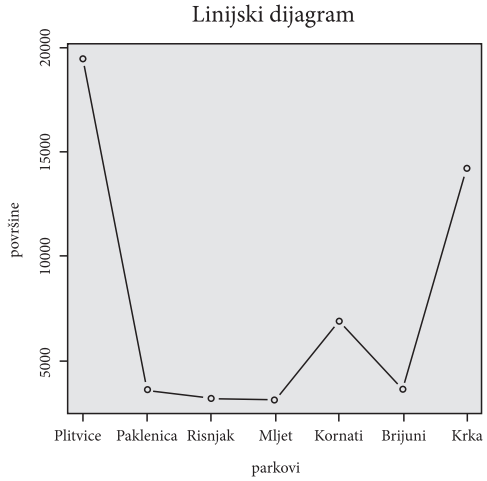
Rješenje.

Podatke za sva tri dijagrama unosimo u obliku vektora stringova *parkovi* s imenima nacionalnih parkova i numeričkog vektora *povrsine* s površinama parkova:

```
> parkovi = c(„Plitvice”,„Paklenica”,„Risnjak”,„Mljet”,„Kornati”,
„Brijuni”,„Krka”)
> povrsine = c(19479,3617,3198,3100,6900,3635,14200)
```

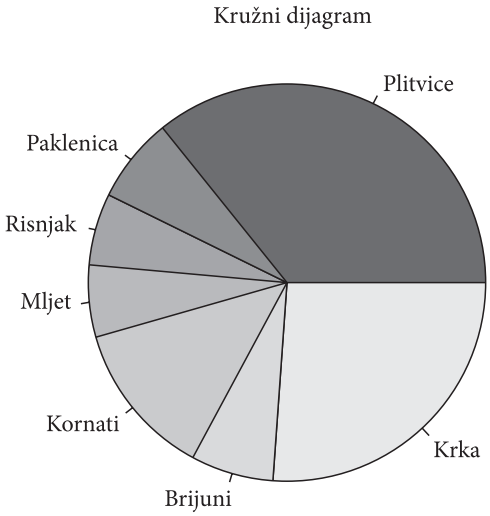
(a) Linijski graf crtamo pozivanjem funkcije *plot* kojoj na prvom mjestu zadajemo vektor s površinama parkova, a ostalim argumentima biramo željene postavke za izgled grafa. Tip grafa postavljamo argumentom *type* na „both”, što znači da će se crtati točke i linije, glavni naslov zadajemo argumentom *main*, ime *x*-osi argumentom *xlab*. Na kraju postavljamo imena točaka na *x*-osi tako da ih prvo poništimo argumentom *xaxt postavljanjem na n*, a onda zadamo s funkcijom *axis*. Funkciji *axis* na prvo mjesto stavljamo redni broj koordinatne osi (npr. 1 za *x*-os), argumentom *at* zadajemo točke umjesto kojih će se upisivati stringovi iz vektora pridruženog argumentu *lab.*

```
> plot(povrsine,type = „b”,main = „Linijski dijagram”,xlab =
„parkovi”, xaxt = „n”)
> axis(1,at = 1:7,lab = parkovi)
```



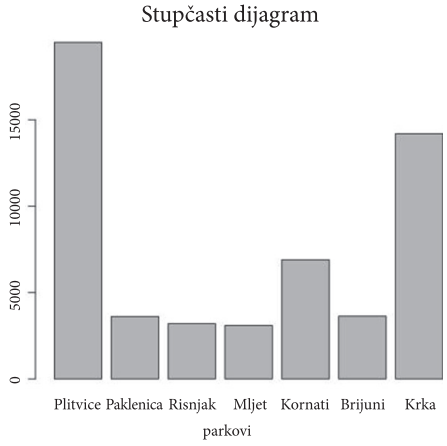
(b) Kružni dijagram crtamo pozivanjem funcije *pie* kojoj na prvom mjestu zadamo vektor s površinama, a argumentu *labels* pridružimo vektor s imenima parkova. S argumentom *main* postavljamo željeni glavni naslov, a s argumentom *col* biramo paletu od 7 nijansi sive boje funkcijom *graycolors*.

```
> pie(povrsine, labels = parkovi, main = „Kružni dijagram”, col
= gray.colors(7))
```



(c) Stupčasti dijagram crtamo pozivanjem funkcije *barplot* kojoj na prvom mjestu zadajemo vektor s površinama. Glavni naslov i ime *x*-osi postavljamo redom argumentima *main* i *xlab*, a imena stupaca argumentom *names.arg*.

```
> barplot(povrsine, main = „Stupčasti dijagram”, xlab = „parkovi”,
names.arg = parkovi)
```



Vodoravni stupčasti dijagram možemo nacrtati dodavanjem argumenta *horiz* postavljenog na *TRUE*.

4. Analiza podataka u *R*-u

R je još jednostavniji kada je u pitanju analiza podataka jer između ostalih ima implementirane sljedeće funkcije: *mean* za aritmetičku sredinu podataka, *var* za varijancu, *sd* za standardnu devijaciju, *median* za računanje medijana podataka, *hist* za crtanje histograma zadanih podataka.

Zadatak 2. Zabilježen je opći uspjeh 30 učenika jednog razreda na kraju godine. Dobiveni su ovi podatci: 4, 3, 4, 3, 1, 3, 4, 3, 3, 3, 2, 4, 1, 5, 3, 4, 1, 3, 3, 1, 3, 5, 4, 3, 1, 4, 5, 4, 1, 3.

(a) Izračunajte aritmetičku sredinu svih ocjena.

(b) Izračunajte varijancu i standardnu devijaciju svih ocjena.

(c) Izračunajte medijan svih ocjena.

(d) Sastavite tablicu frekvencija i relativnih frekvencija.

(e) Dobivene podatke prikažite histogramom.

Rješenje.

(a) Označimo s n ukupan broj podataka. Aritmetičku sredinu dobivamo tako da zbrojimo sve podatke i podijelimo s n. Često je nazivamo srednja ili prosječna vrijednost i označavamo sa:

$$\bar{x} = \frac{4+3+4+3+1+3+4+3+3+3+2+4+1+5+3+4+1+3+3+1+3+5+4+3+1+4+5+4+1+3}{30}$$

$$= 3.033$$

Aritmetičku sredinu možemo u *R*-u izračunati na ovaj način:

```
>   ocjene=c(4,3,4,3,1,3,4,3,3,3,2,4,1,5,3,4,1,3,3,1,3,5,4,3,
1,4,5,4,1,3)
> mean(ocjene)
[1] 3.033333
```

(b) Varijanca uzorka mjeri odstupanja podataka od aritmetičke sredine i računa se kao suma kvadrata razlika podataka i aritmetičke sredine podijeljena s (n-1). Označava se sa:

$$s^2 = \frac{(4-\bar{x})^2 + (3-\bar{x})^2 + (4-\bar{x})^2 + (3-\bar{x})^2 + (1-\bar{x})^2 + \cdots + (3-\bar{x})^2}{29} = 1.55.$$

Izračunajmo varijancu u *R*-u:

```
> var(ocjene)
[1] 1.550575
```

Standardna devijacija je drugi korijen iz varijance pa ona iznosi $s = \sqrt{1.55} = 1.24$, a u *R*-u se može dobiti pozivanjem funkcije sd:

```
> sd(ocjene)
[1] 1.245221
```

(c) Medijan sortiranog niza podataka srednja je vrijednost u slučaju da je *n* neparan broj ili aritmetička sredina dviju srednjih vrijednosti kada je *n* paran broj.

Sortiranjem zadanih ocjena dobivamo ovaj niz:

1, 1, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5.

Kako je ukupan broj ocjena jednak 30, pogledamo koje su ocjene na 15. i 16. mjestu, zbojimo ih i podijelimo brojem 2 pa za medijan dobivamo m = (3 + 3) / 2 = 3.

U *R*-u se medijan dobije pozivanjem funkcije *median*:

```
> median(ocjene)
[1] 3
```

(d) Gledanjem sortiranog niza podataka vidimo da se ocjena 1 pojavljuje 6 puta pa je njena frekvencija 6. Podijelimo li tu frekvenciju s *n*, dobit ćemo da je relativna frekvencija ocjene 1 jednaka 0.2. Analogno napravimo za ostale ocjene pa dobijemo sljedeću tablicu:

Ocjena	Frekvencija	Relativna frekvencija
1	6	6/30 = 0.2
2	1	1/30 = 0.033
3	12	12/30 = 0.4
4	8	8/30 = 0.267
5	3	3/30 = 0.1

Možemo provjeriti rezultat tako da zbrojimo relativne frekvencije. Kako je suma frekvencija jednaka *n*, suma relativnih frekvencija treba biti jednaka 1.

Do frekvencija zadanog niza podataka u *R*-u se dolazi pomoću funkcije *table*:

```
> frekvencije = table(ocjene)
> frekvencije
```

ocjene

1 2 3 4 5

6 1 12 8 3

Na taj se način dobije vektor frekvencija koji u prvom retku ima ispisana imena stupaca, u ovom slučaju sve vrijednosti ocjena od 1 do 5.

Sada možemo relativne frekvencije dobiti dijeljenjem s *n,* što se u *R*-u radi po mjestima tako da se cijeli vektor podijeli s *n*:

```
> n = length(ocjene)
> relativne_frekvencije = frekvencije/n
> relativne_frekvencije
```

ocjene

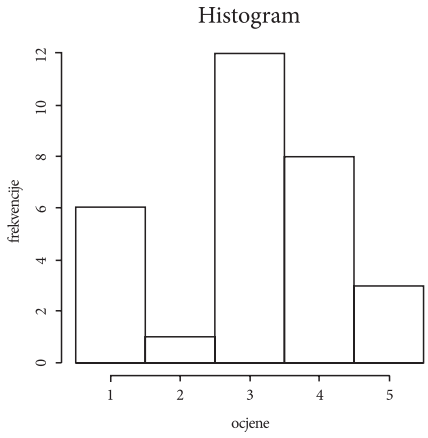
1 2 3 4 5

0.20000000 0.03333333 0.40000000 0.26666667 0.10000000

I za kraj provjerimo je li suma relativnih frekvencija jednaka jedan:

```
> sum(relativna_frekvencija)
[1] 1
```

(e) Prije crtanja histograma potrebno je odrediti granice razreda. Za diskretne podatke uobičajeno je da njihove vrijednosti budu u sredini razreda histograma. Kako je širina stupića histograma jednaka jedan i stupići se međusobno diraju, granice histograma u ovom primjeru trebaju biti redom 0.5, 1.5, 2.5, 3.5, 4.5 i 5.5. Histogram u *R*-u crtamo pomoću funkcije *hist* kojoj na prvom mjestu zadajemo vektor s ocjenama, a argumentom *breaks* biramo niz željenih granica razreda kojih treba biti za jedna više nego što ima različitih vrijednosti podataka. Glavni naslov i ime y-osi zadaju se argumentima *main* i *ylab*.




```
> hist(ocjene, breaks = c(0.5,1.5,2.5,3.5,4.5,5.5), main =
„Histogram”, ylab = „frekvencije”)
```

Histogram relativnih frekvencija u kojemu je ukupna površina stupića jednaka jedan dobivamo tako da unutar funkcije *hist* postavimo argument *prob* (ili *probability*) na *TRUE*:

```
> hist(ocjene, breaks = c(0.5,1.5,2.5,3.5,4.5,5.5), main =
„Histogram", ylab = „relativne frekvencije", prob = TRUE)
```

Dobiveni histogram razlikovat će se od prethodnog samo u skali na *y*-osi.

Zadatak 3. U jednoj meteorološkoj stanici mjerena je temperatura zraka u određenom razdoblju. Dobiveni su sljedeći podatci u °C:

28, 22, 21, 27, 20, 31, 30, 30, 33, 32,
20, 21, 25, 24, 26, 26, 22, 20, 27, 29,
31, 30, 34, 34, 30, 28, 26, 20, 26, 24,
20, 21, 27, 31, 26, 26, 28, 24, 25, 22.

(a) Sastavite tablicu frekvencija i relativnih frekvencija s razredima širine 3.

(b) Nacrtajte histogram frekvencija.

Rješenje.

(a) Različitih vrijednosti mogućih temperatura zraka ima previše da bi se za svaku u histogramu crtao poseban stupić pa ih trebamo razvrstati u grupe, odnosno razrede. Odredimo prvo potreban broj razreda zadane širine 3. Najmanja izmjerena vrijednost je 20, a najveća 34. Stoga je raspon podataka jednak 34 – 10 = 14. Za ukupan broj razreda podijelimo raspon podataka sa željenom širinom 3, što u ovom primjeru daje 4.66 pa zaokruživanjem na prvi veći prirodan broj dobivamo da je za histogram potrebno 5 razreda. Granice razreda treba izabrati tako da imaju jedno decimalno mjesto više u odnosu na podatke zbog jednoznačnog svrstavanja podataka u razrede. Kako razredi trebaju obuhvaćati sve izmjerene temperature, za prvu granicu biramo broj manji od 20. Možemo početi na primjer s 19.5 i dodavati širinu 3 tako da je jedan odabir granica razreda: 19.5, 22.5, 25.5, 28.5, 31.5 i 34.5. Sada smo spremni za računanje frekvencija. Potrebno je prebrojiti koliko podataka ima u pojedinom razredu, a za relativne frekvencije dobivene brojeve podijeliti ukupnim brojem podataka:

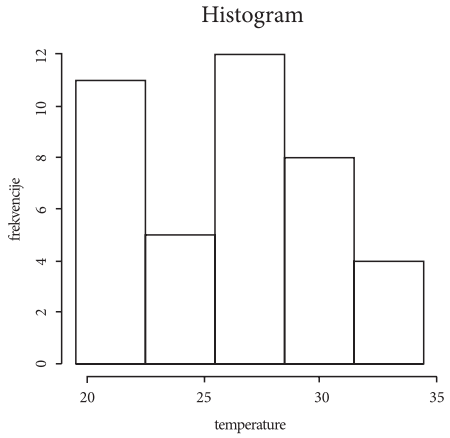
razred	frekvencija	relativna frekvencija
19.5 – 22.5	11	11/40 = 0.275
22.5 – 25.5	5	5/40 = 0.125
25.5 – 28.5	12	12/40 = 0.3
28.5 – 31.5	8	8/40 = 0.2
31.5 – 34.5	4	4/40 = 0.1

Raspon podataka možemo provjeriti u *R*-u na ovaj način:

```
> temperature = c(28, 22, 21, 27, 20, 31, 30, 30, 33, 32,
+ 20, 21, 25, 24, 26, 26, 22, 20, 27, 29,
+ 31, 30, 34, 34, 30, 28, 26, 20, 26, 24,
+ 20, 21, 27, 31, 26, 26, 28, 24, 25, 22)
> max(temperature)-min(temperature)
[1] 14
```

(c) Crtanje histograma sa zadanim razredima u *R*-u ide naredbom *hist* kojoj na prvom mjestu zadajemo vektor s temperaturama, a argumentom *breaks* postavljamo željene granice izračunate u (a) dijelu zadatka. Glavni naslov i naslov y-osi zadajemo argumentima *main* i *ylab*:

```
> histogram = hist(temperature,breaks = c(19.5,22.5,25.5,28.5,
31.5,34.5),main = „Histogram”, ylab = „frekvencije”)
```



Detalje histograma možemo izvući ispisivanjem varijable *histogram* kojoj smo pridružili prethodnu naredbu:

```
> histogram
$breaks
[1] 19.5 22.5 25.5 28.5 31.5 34.5

$counts
[1] 11  5 12  8  4

$density
[1] 0.09166667 0.04166667 0.10000000 0.06666667 0.03333333

$mids
[1] 21 24 27 30 33
```

```
$xname
[1] „temperature”

$equidist
[1] TRUE

attr(,„class”)
[1] „histogram”
```

Na ovaj se način dobiju, između ostalog, granice razreda histograma (*breaks*), frekvencije (*counts*), sredine razreda (*mids*) i ime *x*-osi (*xname*). Ako, na primjer, želimo izvući samo vektor s frekvencijama, dovoljno je nakon imena *histogram* staviti *$counts*:

```
> histogram$counts
[1] 11  5 12  8  4
```

Kod crtanja histograma kod kojeg je površina svih stupića jednaka 1, visine stupića dobivamo tako da relativne frekvencije podijelimo sa širinom razreda (budući da širina razreda neće biti uvijek jednaka 1, kao što je slučaj kod diskretnih podataka). Na primjer, visina prvog stupića bit će jednaka prvoj relativnoj frekvenciji 0.275 podijeljenoj sa zadanom širinom razreda 3, dakle 0.0916. Visine stupića histograma spremljene su u vektoru s imenom *density* pa ih u *R*-u možemo ispisati ovako:

```
> histogram$density
[1] 0.09166667 0.04166667 0.10000000 0.06666667 0.03333333
```

5. Rad s postojećim skupovima podataka u *R*-u

R sadrži dosta ugrađenih skupova podataka koji mogu poslužiti kao baza za kreiranje novih primjera za vježbu. Mogu se pregledati naredbom:

```
> data()
```

Jedan od njih je „islands”, skup podataka koji sadrži površine najvećih kopnenih masa na Zemlji. Površine su zapisane u tisućama kvadratnih metara. Više informacija o ovom skupu podataka može se dobiti pomoću:

```
> ?islands
```

Prije korištenja je skupove podataka potrebno učitati u *R* navođenjem njihovog imena unutar funkcije *data*:

```
> data(„islands”)
```

Izračunajmo sada redom njegovu duljinu, maksimalnu, minimalnu i prosječnu površinu:

```
> length(islands)
[1] 48
> max(islands)
[1] 16988
> min(islands)
[1] 12
> mean(islands)
[1] 1252.729
```

Da bismo dobili još bolji osjećaj o podatcima, možemo pokušati nacrtati neke od već spomenutih grafičkih prikaza i uočiti da dobivene slike nisu baš zadovoljavajuće zbog velikog broja podataka na *x*-osi. U takvim slučajevima kao dobra zamjena stupčastog dijagrama može poslužiti točkasti dijagram u kojemu se podatci prikazuju horizontalno, a poziva se naredbom *dotchart*:

```
> dotchart(islands)
```

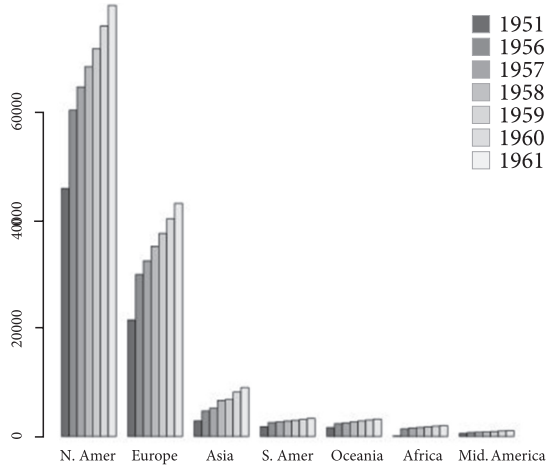
Za kraj, pogledajmo skup podataka „WorldPhones" koji sadrži ukupan broj telefona u raznim područjima svijeta (u tisućama) za 7 različitih godina, učitajmo ga i spremimo u varijablu *telefoni*:

```
> data(„WorldPhones")
> telefoni = WorldPhones
> telefoni
```

	N.Amer	*Europe*	*Asia*	*S.Amer*	*Oceania*	*Africa*	*Mid.Amer*
1951	*45939*	*21574*	*2876*	*1815*	*1646*	*89*	*555*
1956	*60423*	*29990*	*4708*	*2568*	*2366*	*1411*	*733*
1957	*64721*	*32510*	*5230*	*2695*	*2526*	*1546*	*773*
1958	*68484*	*35218*	*6662*	*2845*	*2691*	*1663*	*836*
1959	*71799*	*37598*	*6856*	*3000*	*2868*	*1769*	*911*
1960	*76036*	*40341*	*8220*	*3145*	*3054*	*1905*	*1008*
1961	*79831*	*43173*	*9053*	*3338*	*3224*	*2005*	*1076*

Želimo li stupčastim dijagramom usporediti promjenu broja telefona za sva područja i godine istovremeno, potrebno je unutar funkcije *barplot* dodati argument *beside* postavljen na *TRUE*. Da prikaz bude još jasniji, možemo dodati i legendu tako da argumentu *legend* pridužimo vektor s imenima godina, odnosno imenima redaka danog skupa podataka koji se u *R*-u dobiva pozivanjem funkcije *rownames*:

```
> barplot(telefoni, beside = TRUE, legend = rownames(telefoni))
```



Na ovoj adresi moguće je pogledati još jednostavnih primjera izračunatih na skupovima podataka iz *R*-a: http://www.r-tutor.com/elementary-statistics.

Zaključak

Iako su u članku birani samo osnovni primjeri koji mogu biti dio srednjoškolske nastave i cilj je bio što jednostavnije zapisati naredbe potrebne da se oni riješe u *R*-u, moguće je dobiti osjećaj koliko je ovaj programski jezik jednostavan za upotrebu. Njegova prednost je svakako i to što se može koristiti i distribuirati besplatno. Svoj razvoj *R* može zahvaliti internacionalnoj ekipi čiji članovi surađuju preko interneta, kao i velikom broju korisnika po cijelom svijetu koji pridonose popularnosti i funkcionalnosti *R*-a razvojem novih programa. S druge strane, činjenica da je temeljen na formalnom kompjutorskom jeziku daje mu veliku fleksibilnost, posebno kada je u pitanju kompleksnija statistička analiza. Zbog svih navedenih razloga, *R* je prikladan alat kod učenja osnova statistike jer nije prekompliciran za korištenje, a može odlično poslužiti za računanje i crtanje grafova i u drugim granama matematike. Primjena programskog jezika *R*-a u nastavi gradiva iz matematike (kao što je grafički prikaz i analiza podataka) može potaknuti bolje razumijevanje i veću zainteresiranost te istovremeno poslužiti kao uvod i vježba za korištenje drugih programskih jezika.

Literatura

1. Daalgard, Peter: Introductory Statistics with R, Springer, 2008.
2. Verzani, John: SimpleR – Using R for Introductory Statistics (elektoničko izdanje javno je dostupno na adresi: https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf)
3. Spiegel, Murray R. i Stephens, Larry J.: Schaum's Outline of Theory and Problems of Statistics, New York: McGraw-Hill, 1999.
4. Vranjković, Petar: Zbirka zadataka iz vjerojatnosti i statistike, ŠK, Zagreb, 1991.
5. Materijali dostupni na internetskoj stranici: http://web.math.pmf.unizg.hr/nastava/stat/files/vjezbe_novo.pdf